

第1次練習-練習-PC1

學號：112111209

姓名：張香裕

作業撰寫時間：40 (mins, 包含程式撰寫時間)

最後撰寫文件日期：2024/12/28

本份文件包含以下主題：(至少需下面兩項，若是有多者可以自行新增)

- ☒ 說明內容
- ☒ 個人認為完成作業須具備觀念

說明程式與內容

開始寫說明，該說明需說明想法，並於之後再對上述想法的每一部分將程式進一步進行展現，若需引用程式區則使用下面方法，若為.cs檔內程式除了於敘述中需註明檔案名稱外，還需使用語法```語言種類 程式碼```，其中語言種類若是要用python則使用py，java則使用java，C/C++則使用cpp，下段程式碼為語言種類選擇csharp使用後結果：

```
public void mt_getResult(){  
    ...  
}
```

若要於內文中標示部分網頁檔，則使用以下標籤```html 程式碼```，下段程式碼則為使用後結果：

```
<%@ Page Language="C#" AutoEventWireup="true" ...>  
  
<!DOCTYPE html>  
  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head runat="server">  
<meta http-equiv="Content-Type" ...>  
    <title></title>  
</head>  
<body>  
    <form id="form1" runat="server">  
        <div>  
            </div>  
    </form>  
</body>  
</html>
```

更多markdown方法可參閱<https://ithelp.ithome.com.tw/articles/10203758>

請在撰寫"說明程式與內容"該塊內容，請把原該塊內上述敘述刪除，該塊上述內容只是用來指引該怎麼撰寫內容。

1. 請執行下面Git操作 (請參照題目pdf)

Ans:

2. 請整理課堂上所提的Git指令，顯示其語法語給予明顯的例子

Ans:

add(新增)

範例: `git add [<檔案名稱.副檔名> 或 <.> <--all> 新增全部]`

結果: 將檔案加入索引

commit(提交)

範例: `git commit`

結果: 將檔案提交上成為一個新版本

push(推送)

範例: `git push`

結果: 想當然提交完版本後就是上傳，將版本從本地推送至GitHub上

fetch(獲取)

範例: `git fetch`

結果: 從GitHub上下載所有新的資料

pull(拉取)

範例: `git pull`

結果: 將伺服器端最新的倉庫分支(branch)與客戶端的進行對應分支合併(merge)

branch(分支)

範例: `git branch (分支名稱)`

結果: 創建新分支

checkout(切換)

範例: `git checkout (分支名稱)`

結果: 切換至所選分支

merge(合併)

範例: `git merge (分支名稱)`

結果: 將所選分支名稱併入當下所處分支

3. 請先寫出同學認為資料結構一門課每周需要花多少時間？並提出理由。並將「請問要學習大學的資料結構每一週需要對於一般大學生需要多少時間」貼入chatgpt，並將其所建議的敘述貼出後，比較同學所認為若是每周低於chatgpt所評估的時間，兩者主要差異為何？應該怎麼安排自己時間了解資料結構？

Ans:

個人認為資料結構課程一周需花3-5小時來處理課程資訊，並且視作業情況再延伸2-4小時製作，但也可能需要花更多時間分析課程內容，如較複雜的時間複雜度。

>GPT回復：

學習大學資料結構課程一般每週需 8-12 小時，包括：

課堂時間：2-4 小時

複習與閱讀：1-2 小時

編程練習：3-5 小時

額外學習：1-2 小時

學習成效取決於背景知識、課程強度與目標。建議每天規律學習 1-2 小時，結合實作與討論，加深理解。

題2：

兩者差異對我來說基本上差不多，課程平均約為4小時，而後續複習、編成、分析總和佔4小時以上。

因應個人程度及學習成效去做變化，先學習再分析，分析後要確認自己所學內容的概念，再去做更深入的操作才能完整學習一項技能。

個人認為完成作業須具備觀念

開始寫說明，需要說明本次練習需學會那些觀念 (需寫成文章，需最少50字，並且文內不得有你、我、他三種文字)且必須提供完整與練習相關過程的notion筆記連結